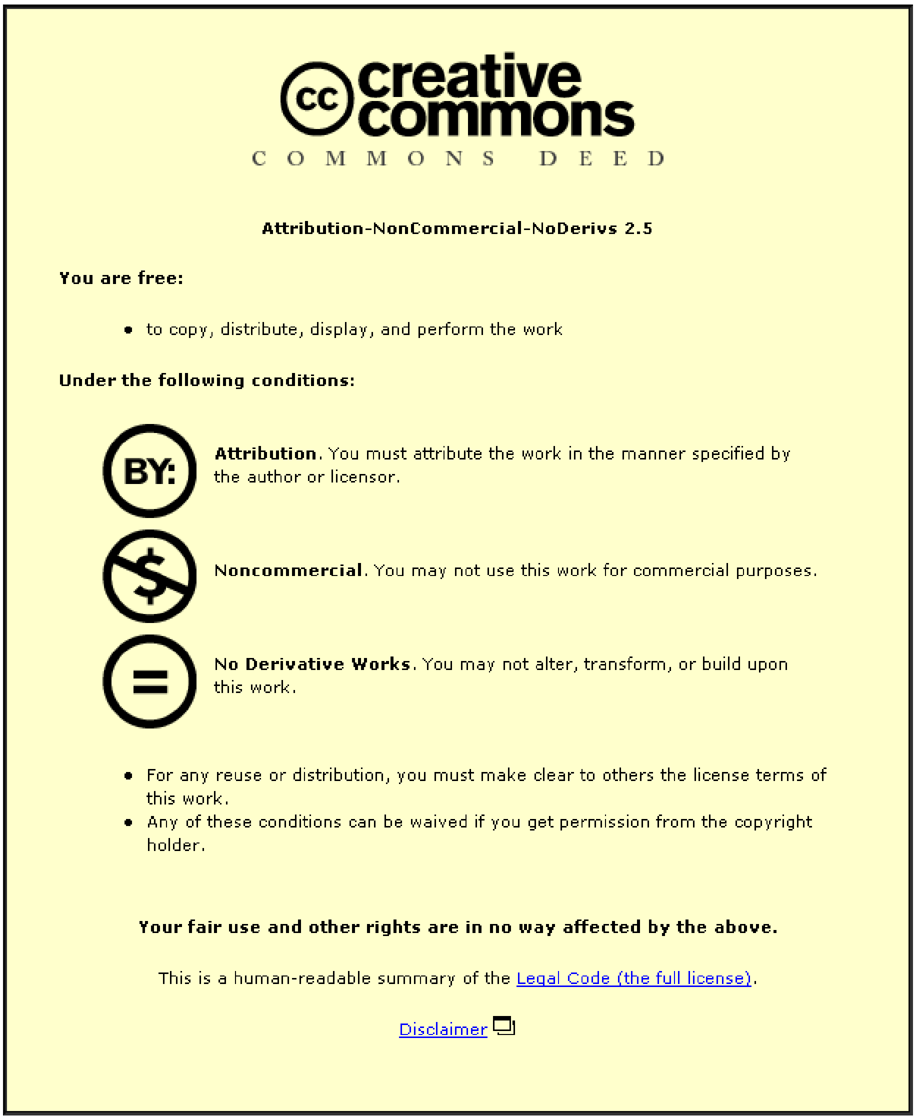

DESIGNING NEURAL NETWORKS FOR MANUFACTURING PROCESS CONTROL SYSTEMS

C. H. Messom and C. J. Hinde,
Department of Computer Studies,
and
A. A. West and D. J. Williams,
Department of Manufacturing Engineering,
Loughborough University of Technology,
Loughborough, Leicestershire, LE11 3TU, UK.

The problem of designing neural networks for the control of discrete manufacturing processes is addressed in this paper. Rather than treating the networks as adaptive black boxes, a new architecture is introduced that links the weights associated with the nodes and thus allows the relationships and internal structure to be tightly constrained. The constrained search space gives greater confidence in the internal representations that have been induced by the training set and therefore about the correct behaviour of the network between the given limits. The method is illustrated by its application to the dispensing of adhesives.

Introduction

The solution of typical problems encountered in manufacturing process control are examined in this paper. These problems can be generally described as maintaining the process output between set limits and identifying the actions to be taken when the output approaches or exceeds the limiting conditions. Limits are reached either because of process drift or because of catastrophic process errors. These characteristics are present in many manufacturing processes, for example the dispensing of adhesives in the manufacture of "mixed technology" printed circuit boards (PCB's) ([1], see below) and metal cutting [14]. The processes can be characterised as being partially understood. Some characteristics can be described and modelled whereas others, such as the interactions between various process variables, cannot.

Representations that allow controllers to be informed of certain of the process characteristics and learn the balance are of importance to the control of such processes. Neural networks provide an attractive method of learning the unknown characteristics [3]. They are, however, difficult to configure and interpret after teaching (see [7] and [11]). The ease with which the internal representations of neural networks can be interpreted is important, especially when such controllers are to be applied in safety critical applications. Hybrid networks designed using readily interpretable models therefore have considerable advantages.

This paper outlines the design of a novel neural network architecture based upon linking the characteristics of several nodes. The application of the network architecture to the accurate dispensing of small quantities of adhesives is then described. The network architecture is particularly attractive for this application because it allows well understood system level control conditions to be rapidly included ([5] and [2]), the relationships between these to be learned and the system to be rapidly modified when the material to be dispensed is changed.

Neural Network Techniques

Several Properties of neural networks make them ideal for application in control processes. Their linear threshold properties allow for immediate quantisation of the system variables, while their Boolean transformation property allows for their use in intelligent control systems.

These two properties are discussed further. It is shown that a network making use of linked parallel nodes is ideal for modelling control problems and indeed is essential if the behaviour of the network is to be fully understood.

Simplified Single System Variable Case

Nodes in a neural network are linear threshold devices. A node with a single input is a particularly simple threshold unit (see figure 2). Figure 1 shows a visual representation of the threshold operation.

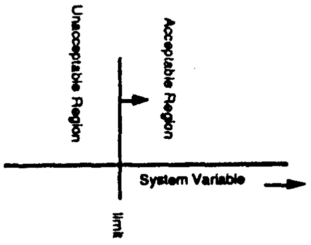

Figure 1. Characterising a single variable with a single split for acceptable and unacceptable decisions.

The discriminating surface is a single point given by the equation:

$$\text{Bias} + \text{Weight} * \text{Input\_Variable} = 0$$

The output is 1 if the left hand side is greater than 0 and -1 if it is less than 0, the cross over point is defined by the equation.

Figure 3 shows the relevant algorithm.

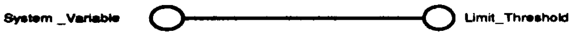

Figure 2. The neural net implementation of a single threshold.

The single threshold can be implemented in a neural network with a single unit as shown in figure 2. The definition of the boundary in one dimension merely specifies a point on the System_Variable axis which satisfies the equation: System_Variable - limit = 0.

By setting the Bias_Weight to - limit (minus limit) and the Input_weight to 1.0, if the value of the System_Variable is greater than the limit then the negative contribution of the Bias_Weight is overcome by the value of the System_Variable and the output is deemed acceptable.

```
Acceptable
      if
            Bias_weight +
            Input_weight *System_Variable
      > 0
Not_Acceptable
      if
            Bias_weight +
            Input_weight*System_Variable
      < 0

 Bias_weight =     - limit
 Input_weight = 1.0
```

Figure 3. The equations and weights which could be assigned to the network shown in figure 4 to implement the decision on a single system variable.

Segmentation of a Region

An important element of control processes is to initiate action in a specified region of the system control variables. The neural network systems are ideal for segmenting areas of the system variables.

The segmentation for a single system variable is shown in figure 4. This cannot be implemented in a single layer network and requires a two layer network as shown in figure 5. The inputs to the final single node are in the open interval (-1,1) with the majority of values close to the end points. If we employ a simple threshold system we are taking values from set {-1,1} and can treat any subsequent layers as implementing Boolean transformations.

The resulting network then has two parts, the first layer taking the real valued input values and separating them into regions, the second layer then combining these regions. The output node implements an OR transformation.

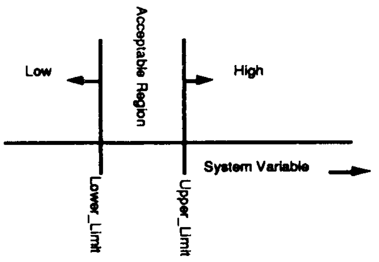

Figure 4. Splitting the system variable into three regions, unacceptably low, acceptable and unacceptably high.

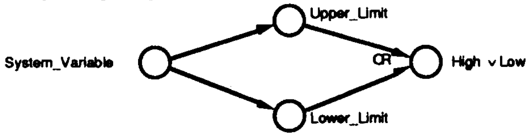

Figure 5. The neural net implementation of a single system variable that must be kept within limits.

Upper_Limit and Lower_Limit segment the system variable into three parallel regions (see figure 4 and the equations in figure 6). The corresponding nodes denoted Upper_Limit and Lower_Limit defined by the equations below (figure 6,7 and 8), are parallel, since their weight values are linearly dependent.

```
Upper_Limit - 1.0*System_Variable = 0
-Lower_Limit + 1.0*System_Variable = 0
```

Figure 6. The equations of the surfaces used to separate the regions in figure 4 implemented in the network shown in figure 5.

```
Acceptable
      if
            - Upper_limit +
            1.0 *System_Variable
      < 0
Not_Acceptable
      if
            - Upper_Limit +
            1.0*System_Variable
      > 0
```

Figure 7. The equations and weights which could be assigned to the network shown in figure 5 for the Upper_Limit node.

```
Acceptable
      if
            Lower_limit +
            -1.0 *System_Variable
      < 0
Not_Acceptable
      if
            Lower_Limit +
            -1.0 *System_Variable
      > 0
```

Figure 8. The equations and weights which could be assigned to the network shown in figure 5 for the Lower_Limit node.

Segmenting Several Regions

Many control problems require several regions of the control variables to be segmented. A simple example of a bimodal warning illustrates this point. Figure 9 shows the two regions where the signal should be active and the three regions where the signal is inactive.

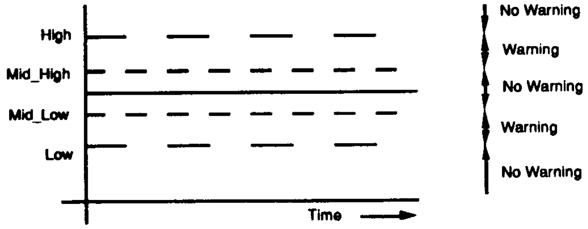

Figure 9. Regions where the inhibited warning node is active and inactive.

The system is not required to give a warning for the case error > High as it will be taking action at this point. This is an inhibited warning. A suitable network architecture implementing bimodal warning outputs, making use of the parallel segmentation nodes, is shown in figure 10. All the nodes, High, Low, Mid_High, Mid_Low are parallel, since they segment a single process control variable.

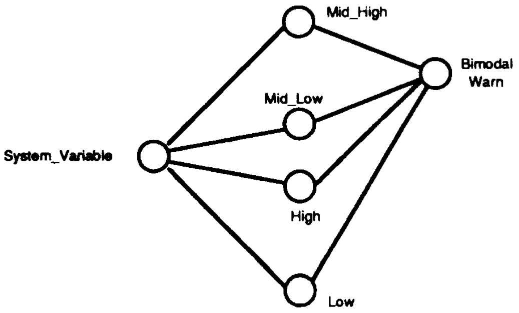

Figure 10. A network structure for implementing the bimodal warning node.

Multiple System Variables

Many control processes are possible when considering multiple system variables. The variables may interact in nonlinear algebraic ways which requires system investigation to discover the relevant combination of variables. Since implementation of control processes when many control variables are present is a difficult design task, a suitable methodology is presented. Figure 11, shows the three main components relevant for designing a neural network control process.

System Investigation

All complex processes will be influenced by several system variables. Once the relevant process variables have been discovered, the algebraic relationships between them must be discovered.

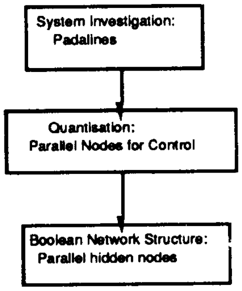
System Investigation: Padalines
Quantisation: Parallel Nodes for Control
Boolean Network Structure: Parallel hidden nodes

Figure 11. Methodology for designing neural network controllers of complex processes.

This is achieved by an investigation of the relevance of the algebraic combinations of process variables. This technique has been used before ([4], [12]) and is derived from a combination of polynomial (p) and adaptive linear neuron (adaline), they are known as polynomial adaptive linear neuron (padaline). The basic idea is that new input nodes can be formed from functions of the basic input nodes, if the function being modelled is the sum of several elementary functions then the padaline can discover the coefficients of the separating surface.

```
Acceptable
 if
   Bias_weight +
   $\sum_i$Input_weight(i) *System_Variable(i)
   > 0
Not_Acceptable
 if
   Bias_weight +
   $\sum_i$ Input_weight(i)*System_Variable(i)
   < 0
```

Figure 12. Algorithm that defines a threshold node with many inputs.

Quantisation of the Inputs

Once the relevant system variables have been discovered by the padaline layer, the quantisation layer acts as either a linear thresholding layer, see Figure 12, or a segmentation layer.

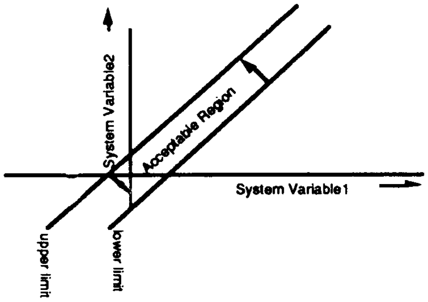

Figure 13. Segmentation of a two control variable system.

A pair of parallel opposite facing nodes in the quantisation layer can segment the control variables into three regions. See figure 13 for a visual representation of the regions when only two system variables are present. Similarly more regions can be segmented with the use of more parallel nodes.

The Boolean Network Structure

The final part of the network is the Boolean transformation that is applied to the the selected region. All control problems require specific action to be taken when the process system is in a specific state. The controller has to provide suitable outputs when the inputs are within a specific region. This is the case however complex the relationship between the system variables. This is shown for a specific two variable case in figure 14. The process should follow the ideal path to within a given tolerance. If the process is within this region no action is taken, while for all other regions corrective action must be taken.

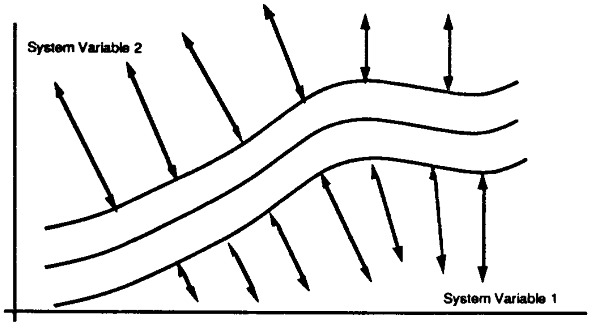

Figure 14. Process with two system variables showing the ideal path to within a given 'olerance.

The selection of these regions with suitable decisions can be made in just one layer, (refer to figure 10), since the multivariable case is analogous to the single variable case. If we are implementing an intelligent controller that requires different remedial action given different segments of the input space a more general two layer network must be used.

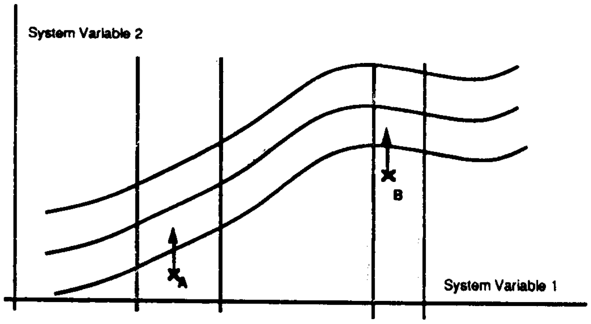

Figure 15. Process with two system variables showing two points that require distinct corrective actions.

Referring to figure 15, the point A and the point B require corrective action to restore the process to the desired region. However, the action that must be applied is different in each case since they lie in different segments of System Variable 1. The Boolean transformation required to implement this decision requires two layers [9].

Standard Control Net

Beyond the first layer of a network the values input to and output from a node are in the set {-1,1} and therefore all transformations beyond that layer are Boolean. The output of any node is a Boolean value and so all the quantisation must occur in the first layer. This results in real values being turned into ranges and so is referred to as the quantisation layer.

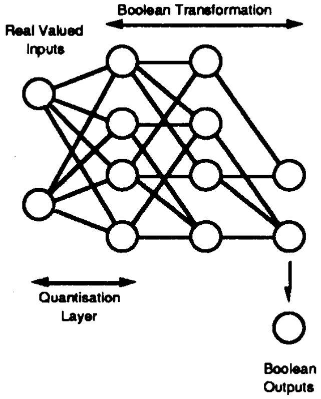

Figure 16. The control network architecture with real valued inputs and interacting quantisations.

Figure 16 illustrates a composite network showing an initial fully connected quantisation layer followed by a Boolean transformation layer. This would be the form of a standard control network that does not exploit the parallelism inherent in control processes.

Parallel Nodes in Neural Network Control Systems

Simple independent quantisation that segments the control parameters provides sets of parallel nodes (the quantising nodes themselves). Transformed control parameters, e.g padalines, that are quantised (figure 16) with parallel regions in higher dimensional spaces also provide parallel nodes. Therefore, parallel nodes occur naturally and are the ideal representational scheme, whether we are considering simple quantisation or interconnected quantisation. The Boolean transformation layer can similarly be implemented by parallel nodes (see [8] and [9]). The complete network structure from the quantisation layer to the Boolean transformation layers can be naturally implemented using parallel nodes.

The network structures with these specified relationships has much fewer weight values to determine, most of them being interrelated. There is more knowledge available about the topology and internal representations of the network since these are made explicit by the parallel relationships. Having more knowledge about the network allows the understanding and interpretation of the representation to be more successful.

Architecture of the Loughborough Control Net

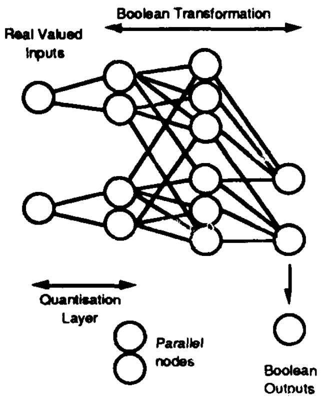

Figure 17. A Loughborough control net. The architecture explicitly shows the nodes that are parallel.

The Loughborough Control Net makes use of the property that parallel nodes can model most control processes, by structuring the architecture of the net so that the parallel nodes are explicitly shown. Figure 17 illustrates this. The connection pattern is dependent on the control process that is modelled. It is constructed by the analysis of the control problem and by training on suitable training points.

Understanding Network Behaviour

One problem of neural networks which must preclude their widespread use in safety critical areas is the lack of knowledge of the internal models that have been formed in the net. If an unstructured network (see figure 16) correctly represents the training set, its behaviour over this set is perfectly predictable. This can not be said about the remaining points in the space of system variables. The actual transformation that the network structure implements is difficult to discover and so, its behaviour is hard to predict. Making use of the parallel dependencies helps to alleviate this problem. It allows the control network's behaviour to predicted perfectly over the total space of system variables. Therefore the behaviour of a network can be programmed given a suitable methodology and its behaviour fully understood. No part of the networks behaviour is undefined.

A network can be engineered to have a particular constrained topology and weight set which can then be tuned appropriately to provide the required behaviour. All the weights and biases will be interrelated. This means that understanding the function of one node will aid the understanding of the corresponding parallel node. This is also the case even if the network has been trained on sample data.

Two parallel and opposite nodes effectively make no contribution in output to the region that is not between the two nodes. This is seen from the fact that the contribution from the first node, "+1" say, and that of the second node, "-1", effectively cancel when the contribution are added by a node in the next layer. (See [8] for further discussion). Throughout this discussion pairs of discriminating nodes that segment a region are parallel and opposing. Therefore the effect of the two parallel nodes on any point outside the enclosed region is cancelled. The enclosed region is interpreted as an area of the input space where a specific output is given. The behaviour of the complete network structure is easily understood, since the parallel planes that form the hidden layers can be interpreted independently.

The Application

To illustrate the application of the network architecture to a real process, the characteristics of the dispensing of adhesives in the manufacture of mixed technology PCB's are outlined below. Earlier work into neural network based control of the adhesive dispensing system [13] addressed the question of the feasibility of this type of controller for this particular process. Since the input data to the network was preprocessed, much of the problem solving lay outside the network. Furthermore there was no a priori design of the topology and internal representations: a fully interconnected network trained with appropriate data was modified in an ad hoc manner until satisfactory convergence was obtained.

In the manufacture of mixed technology PCB's, the surface mount components are secured to the board, prior to a wave soldering operation, by a small (0.0002 to 0.005 ccs depending on component) amount of adhesive. The amount of adhesive dispensed is critically dependent upon several process environment variables (e.g. temperature, humidity, erratic thixotropic behaviour of the adhesive, air bubbles in the flow and variations in the PCB substrate).

The dispensing unit consists of a syringe of adhesive coupled to a pressure control unit. The unit is made up of a solenoid valve, pressure regulator, temperature sensor and a pressure transducer to monitor the variation of pressure within the syringe. The dispensing unit is fixed to a SEIKO RT3000 robot which moves the syringe to locations on the PCB where the adhesive has to be dispensed, either for good or bad dispense operations. Feedback data collection is carried out by an image processing system (Imaging Technology ITI 151) coupled to a Pulnix TM-460 CCD camera incorporating a magnifying optical system.

The original system, as reported in [1], was developed using the MUSE real time AI toolkit and used bang-bang rule based control as the main paradigm. MUSE is a hybrid modular system supporting a range of knowledge representation paradigms; PopTalk, a procedural language with object oriented programming extensions, a forward chaining rule language, a backward chaining language, data directed programming through the use of daemons and flexible relation supporting general relations between objects. Particular support for real time operation included agenda based priority scheduling, interrupt handling and fast data capture.

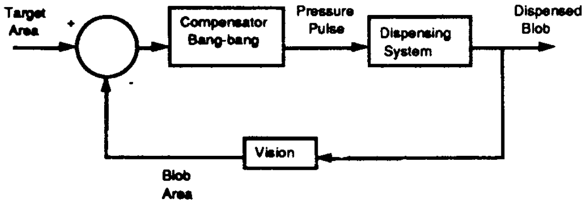

Figure 18. Feedback control loop of the adhesive dispensing system.

Figure 18 illustrates the basic feedback control loop of the adhesive dispensing system. By varying the height and/or width of the pressure pulse applied to the syringe the amount of adhesive dispensed can be controlled. Feedback data consists of the plan area of the dispensed blob. Differences between this and the target size required for a particular component provides the input for the process controller. Steady state control is illustrated in figure 19.

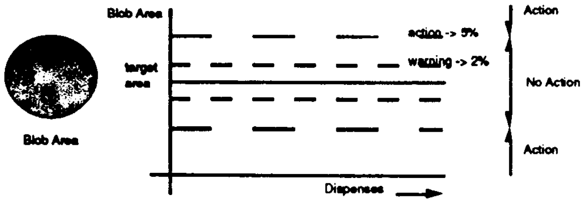

Figure 19. Steady State Control of the size of the adhesive blob.

Control consists of maintaining the measured area within bands around the target. Action is taken whenever the measured area drifts outside the +/- 5%. Figure 10 illustrates the network that can be easily configured to provide control.

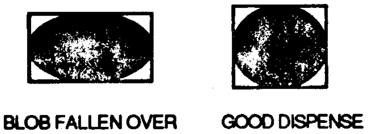

Figure 20. Illustration of good and bad dispenses.

There are several process faults that occur in the dispensing of blobs of adhesive. These faults are generally observed when process variables or trends in process variables exceed heuristically determined thresholds and thus can be monitored using simple networks that implement single thresholds (see figure 2). For example (a) solder pad contamination can occur when an incorrect blob shape is dispensed (figure 20). This is usually due to trailing the adhesive blob as the dispense head is moved to the next location. This "blob fallen over" condition is monitored by determining the ratio of the measured area of the blob to the area of a box that encloses the blob. For a perfect dispense this box area ratio (BAR) is ~0.8. Solder pad contamination occurs at values BAR <0.6 (figure 21).

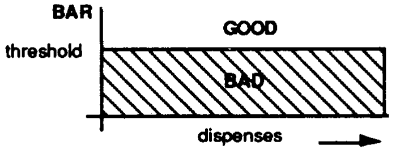

Figure 21. The Box Area Ratio is a good indicator of the quality of an adhesive dispense operation.

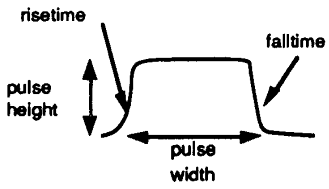

Figure 22. Parametrisation of the variation of pressure within the syringe of adhesive.

(b) The measured variation of pressure within the syringe is shown in figure 22. The pressure pulse is characterised by its risetime, pulse height, pulse width and falltime. Faults are indicated when parameters vary outside their allowable range. For example large increases in the risetime and falltime have been found to be caused by air leaks and sticking solenoid valves respectively.

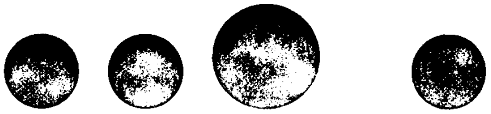

Figure 23. The increasing size of blob can indicate the presence of a bubble.

(c) voids in the flow of adhesive can lead to missed dispenses where a blob is absent. The onset of a "bubble" is characterised by a dramatic increase in the in the size of the adhesive blob in the absence of a corresponding increase in the dispense pressure (figure 23).

A simple neural network solution for the adhesive dispensing system is apparent from the above. Using a series of small independent networks a simple bang-bang controller and error monitor is easily determined.

The interdependence of the individual parameters is currently under investigation. Figure 24 illustrates the methodology adopted (see also figure 11).

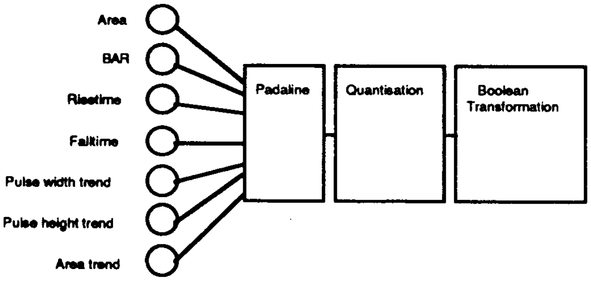

Figure 24. The methodology adopted in determining the interdependencies between the adhesive dispensing parameters.

The interdependence is determined from real data using a padeline (although standard system identification techniques [6] are also under investigation). Quantisation leads to a determination of the various thresholds and the control rules determined in the Boolean part of the network [10].

Conclusions

The paper has addressed the problem of designing neural networks rather than treating them as adaptive black boxes. The use of dependencies in the hidden layers tightly constrains the space of possible configurations of the network. The interdependencies introduced of making the nodes parallel allows us to be more confident about the internal representations that are induced by the training set.

References

[1] R. Chandraker, A. A. West and D. J. Williams, "Intelligent control of Adhesive Dispensing", IJCIM, Special issue in Intelligent Control, vol. 3, No 1,1990, pp. 24-34.

[2] E. Grant and B. Zhang, "A neural net approach to supervised learning of pole balancing", Proc. 4th International Symposium on Intelligent Control, Albany, New York, 25-27 September, 1989.

[3] J. Hertz, A. Krogh and R. G. Palmer, Introduction to the Theory of Neural Computation, Addison Wesley Publishing Company, Wokingham, United Kingdom, 1991.

[4] C. J. Hinde, Heuristic techniques applied to an industrial situation, Ph.D. thesis Brunel University, 1974.

[5] C. J. Hinde, "Some experiments on accelerating the convergence of neural networks", Dept. Computer Studies Internal Report 679, University of Technology Loughborough UK, 1992.

[6] L. Ljung, System Identification: A Theory for the User, Prentice Hall, Englewood Cliffs, N.J., 1987

[7] T. Materna, "Neural networks enter high speed marketplace", Computer Technology Review, vol. 7, No 7, June 1987.

[8] C. H. Messom, "Dependencies in the hidden layer of neural systems", Dept. Computer Studies Internal Report 674, University of Technology Loughborough UK, 1992.

[9] C. H. Messom and C. J. Hinde, "Topological limits on neural nets", Dept. Computer Studies Internal Report 675, University of Technology Loughborough UK, 1992.

[10] C. H. Messom, C. J. Hinde, A. A. West and D. J. Williams, "A Neural Network for the Control of an Adhesive Dispensing System", in preparation.

[11] G. Mirchandani and W. Cao, "On Hidden Nodes for Neural Nets", IEEE Transactions on Circuits and Systems, vol. 36, pp 661- 664, 1989.

[12] D. F. Specht, "Probabilistic neural networks and the polynomial adaline as complementary techniques for classification", IEEE transactions on neural networks, vol. 1, No. 1, pp 111-121, 1990.

[13] A. Valmiki, A. A. West and D. J. Williams, "The evolution of a neural net controller for adhesive dispensing", submitted to the International Federation of Automatic Control Workshop on Computer Software Structures Integrating AI/KBS Systems in Process Control, 1990.

[14] P. K. Wright, E. Pavlakos and F. Hansen, "Controlling the Physics of Machining on a New Open-Architecture Manufacturing System", ASME Winter Annual Meeting, 1991.